Evaluating auction-based task allocation in multi-robot teams

Eric Schneider[1], Ofear Balas[2], A. Tuna Özgelen[3], Elizabeth I. Sklar[1] and Simon Parsons[1]

[1] Dept of Computer Science, University of Liverpool, UK
[2] Dept of Computer & Information Science, CUNY Brooklyn College, USA
[3] Dept of Computer Science, Graduate Center, City University of New York, USA
eric.schneider@liverpool.ac.uk, ofear.balas@gmail.com,
aozgelen@gc.cuny.edu, e.i.sklar@liverpool.ac.uk,
s.d.parsons@liverpool.ac.uk

Abstract. Task allocation is an important topic in multiagent and multi-robot teams. In recent years, there has been much research on the use of auction-based methods to provide a distributed approach to this allocation. Team members bid on tasks based on their local information, and the allocation is based on these bids. The focus of prior work has been on the optimality of the allocation, and has established that auction-based methods perform well in comparison with optimal allocation methods, with the advantage of scaling better. Here we take a different approach, comparing several auction-based methods not on the optimality of the allocation, but on the efficiency of the execution of the allocated tasks. This approach factors in aspects such as the utilization of the team members and the degree to which they interfere with each others' progress, giving a fuller picture of the practical use of auction-based methods.

1 Introduction

This paper is concerned with the *multi-robot routing* problem. This is a task allocation problem in which tasks require robots to reach particular *target* points, and each target must be visited by one robot only. As [1] points out, multi-robot routing is a standard task that is part of problems such as search-and-rescue and demining, and it is a common test domain for robot coordination [2]. The problem is also computationally hard. The number of ways of allocating m target points to n robots is a Stirling number of the second kind:

$$S(m,n) = \frac{1}{n!}\sum_{j=0}^{n}(-1)^{n-j}\binom{m}{n}j^m$$

and the size of the problem quickly defeats attempts to use standard optimization techniques like integer programming [3].

The computational explosion in the multi-robot routing problem has led to researchers looking for more efficient solutions. For example [4] began an interesting line of work in using *auction mechanisms* to allocate the target points in a multi-robot routing problem. This work showed that multi-round auctions, where each round consisted of bids on single tasks, had a performance that was experimentally determined to be close to that of an optimal allocation.

This initial paper was followed by work on the way that rules for determining bids affected the performance of the auction mechanism, establishing bounds on the performance of different rules [5], and empirically comparing the performance of the auctions against an optima allocation under different bidding rules [3]. Later work considered the *sequential single-item auction* [6, 7] as an alternative to *combinatorial auctions* for multi-robot routing, again showing that the total path cost — the total distance travelled by all the robots in the team — is close to optimal. The same authors have subsequently shown that there are advantages to allowing bundling of goods in such auctions, in terms of reducing the number of bids, but at the cost of increasing the total distances travelled by the team.

Overall, we believe that this work shows the power of auction mechanisms in task allocation in this domain, and this paper builds on this prior work. Our extension is to evaluate the effectiveness of auction mechanisms not only in terms of the total distance that the team expects to travel given the task allocation — what we might consider the theoretical cost of executing the tasks — but also in terms of a set of metrics that measure how the tasks are executed in practice. As we explain below, making these measurements can distinguish between allocations that minimise the total distance travelled by the team (as considered in previous work) and the total time taken to complete the tasks. This reveals the trade-offs between auction mechanisms in different scenarios, and allows us to draw conclusions about the applicability of those mechanisms. To our knowledge, this is the first paper to provide this analysis of the practical application of the sequential single-item auction, and to compare its performance against other auction mechanisms on such metrics.

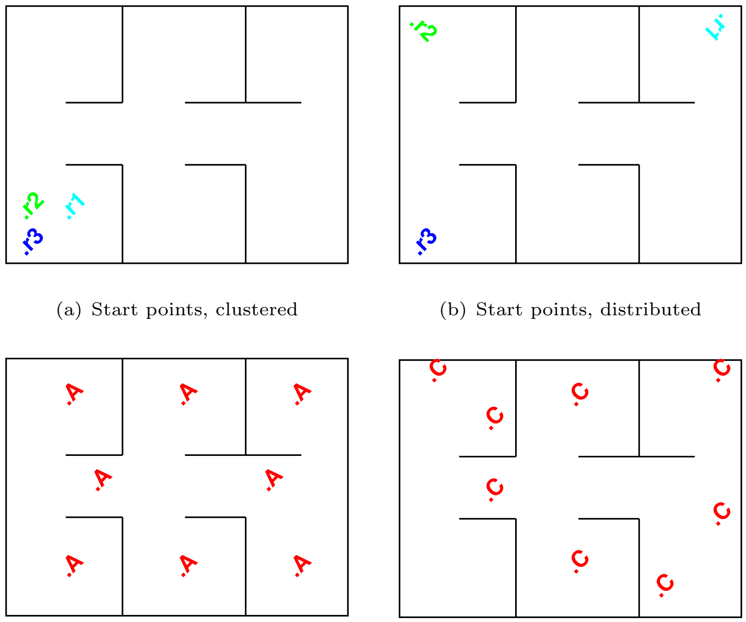
The paper is structured as follows. Section 2 discusses the system that we used to carry out this work and the metrics used to assess the mechanisms we are testing. Section 3 then describes the experiments in detail, gives the results, and provides some analysis. Section 4 discusses the results, Section 5 describes related work, and Section 6 concludes.

2 Methodology

The work reported here is part of a larger effort [8, 9] aimed at evaluating coordination techniques in terms of their potential for application in real robot systems. This influences a number of aspects of the experiments that we conducted. We discuss these aspects in this section.

2.1 System Architecture

Our multi-robot system is fully distributed in the sense that the controllers for the robots are independent agents running as separate processes. No controller



(a) Start points, clustered

(b) Start points, distributed

(c) Target points A

(d) Target points C

Fig. 1. Scenario definitions. The plots on the top show the two sets of starting points superimposed on a map of the test environment. The plots on the bottom show two sets of target points.

has the ability to tell the controller of another robot what to do. The coordination of the team is through a central *auction manager*, which holds a list of target points and communicates the start of an auction, which will award target points to each robot controller[4]. Each controller sends in bids, the auction manager determines the winner of the auction and allocates target points accordingly. Since the individual controllers only have access to their local information, their bids and the allocation don't explicitly attempt to optimize across all robots.

Our software architecture is agnostic about whether the team executes its tasks on real robot hardware or in simulation. That is, the robot controllers are capable of receiving information either from real robots operating in our lab, or from simulated robots operating in a simulation of the lab, and the same decisions are made and the same control signals output in both cases. For the work reported here, we ran the experiments using the Stage simulator [11]. Previous work [9] suggests that for our setup, there is close agreement between simulated results and results obtained on real robot hardware.

[4] Though the bidding process and winner determination are managed centrally, there is no centralized control in the usual sense. The auction could also be distributed among the robots as in [10].

The environment in which our robots, both real and simulated, operate is an office-like environment with rooms opening off a central hallway. The layout of rooms is shown in Figure 1. This is a smaller environment, in terms of the number of rooms, than that studied by [7] (and in similar work by the same group of authors), a fact determined by the maximum size of the physical area available in our lab for the real robots and our desire to be able to run parallel experiments in both simulation and on the real robots.

2.2 Metrics

To measure the performance of the robot team in practice — and hence the coordination mechanisms — we consider a number of metrics applicable to the performance of individual robots and the team as a whole.

In any work with robots, an important consideration is power consumption. This is the fundamental scarce resource that a robot possesses. Robot batteries only last for a limited time, and so, all other things being equal, we prefer solutions to the multi-robot routing problem that minimize battery usage. As in [3–7], therefore, we measure the *distance travelled* by the robots in executing a set of tasks — both individually and as a group — since this is a suitable proxy for power consumption.

It is important to note that distance is not computed by looking at the shortest distances between the target points, but is (as closely as we can establish) the actual distance moved by the robots during task execution. We collect regular position updates from the simulator, compute the Euclidean distance between successive locations, and sum these up. As Figure 2 shows, the resulting paths show plenty of variation from straight line paths.

Distance travelled is not the only metric of interest. In many multi-robot team activities that relate to the routing problem, time to complete a set of tasks is also important. Time is important in exploration tasks — in search and rescue activities, in patrolling, and possibly in demining[5] — and so we also measure *run time*, the time between the start of an experiment and the point at which the last robot on the team completes the tasks allocated to it. Also relevant is the *deliberation time*, the time that it takes for the tasks to be allocated amongst the robots. Deliberation time matters because it feeds into the overall time required to complete a set of tasks, but also because it allows us to establish how different allocation mechanisms compare in terms of the computational effort and communication resources required to run them.

We also measure *idle time*, the amount of time that robots sit idly during the execution of a set of tasks. We compute the idle time for a robot as the time that elapses between when that robot completes its last task and when all robots on the team have completed all of their assigned tasks. This gives us a way of quantifying how equally tasks are distributed among robots, and it also suggests the extent to which resources are being wasted by a particular

[5] One can easily imagine demining happening against the clock — in humanitarian demining [12], for example, there may be the need to demine an area in order to allow refugees to move safely away from a dangerous situation.

allocation. Although the mismatch between the number of tasks and the number of robots means that some idle time is inevitable, idle time represents the use of precious power that is not being directed towards the completion of the tasks.

With several robots moving around a limited space, especially one that requires robots to move through a single corridor to navigate from room to room, robots naturally get in each other's way and are in danger of collisions. We handle this situation using the same approach that was adopted in the first DARPA Grand Challenge [13]. That is, we detect situations where a collision between two robots is likely, stop one robot, and give the other the right of way. Since such episodes impact the other metrics, we count these episodes and record the number of what we call *near collisions*, as well as the amount of time robots spend waiting to give others right of way, which we record as *delay time*.

3 Experiments

In this section we describe a set of experiments that we carried out using the system described above, and give the results of those experiments.

3.1 Experimental Setup

As described earlier, the multi-robot routing problem that we are studying involves a team of robots tasked to visit a number of target points such that one robot visits each point. A *scenario* is a mission defined by a specific set of parameters: the number of robots on the team (n), the starting locations for the robots, the number of target points to visit (m), and the locations of the target points. Thus, a scenario can be described by a tuple

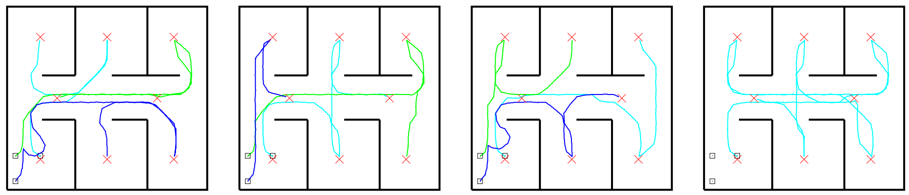
$$\langle n, \{(x_0, y_0), \ldots, (x_{n-1}, y_{n-1})\}, \\ m, \{(x'_0, y'_0), \ldots, (x'_{m-1}, y'_{m-1})\}\rangle$$

The experiments reported here measured results in four different scenarios. All scenarios involved $n = 3$ robots, $m = 8$ target points. There were two sets of starting locations, one distributed and one clustered, and two sets of target point locations, A and C. The starting locations and the target points can be seen in Figure 1. Experiments were conducted with each scenario using each of the four task allocation mechanisms discussed below, and each was run 10 times. Thus 160 experimental trials were conducted in all:

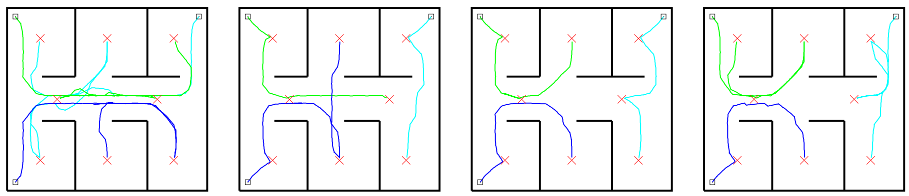
$$2 \textit{ start locations} \times 2 \textit{ target point sets} \\ \times 4 \textit{ allocation mechanisms} \times 10 \textit{ trials}$$

Each experiment recorded the metrics described above:

- Distance travelled;
- Run time;
- Near collisions;



(a) Clustered, A, RR (b) Clustered, A, OSI (c) Clustered, A, SSI (d) Clustered, A, PSI



(e) Distributed, A, RR (f) Distributed, A, OSI (g) Distributed, A, SSI (h) Distributed, A, PSI

Fig. 2. Sample runs on target points A, with start points clustered (top) and distributed (bottom)

– Deliberation time;
– Idle time; and
– Delay time.

From these values we also compute the *travel time* of a robot, which is the run time minus the delay time and the idle time — that is the time that the robot is actually moving.

3.2 Mechanisms tested

Our experiments involved four different mechanisms for task allocation:

– Round-robin
– Ordered single item auction
– Sequential single item auction
– Parallel single item auction

We describe these in turn.

In *round robin* (RR) allocation, we start with two ordered lists, one of target points and one of robots. The first target point is allocated to the first robot, the second interest point to the second robot and so on. When one target point has been allocated to each robot, a second target point is allocated to the first robot. And so on. This is clearly not a particularly efficient way to approach task allocation, but it provides a baseline against which other mechanisms can be tested. We referred to this as the "greedy taxi" policy in our earlier work [9], because this policy emulates the behaviour of a taxi rank.

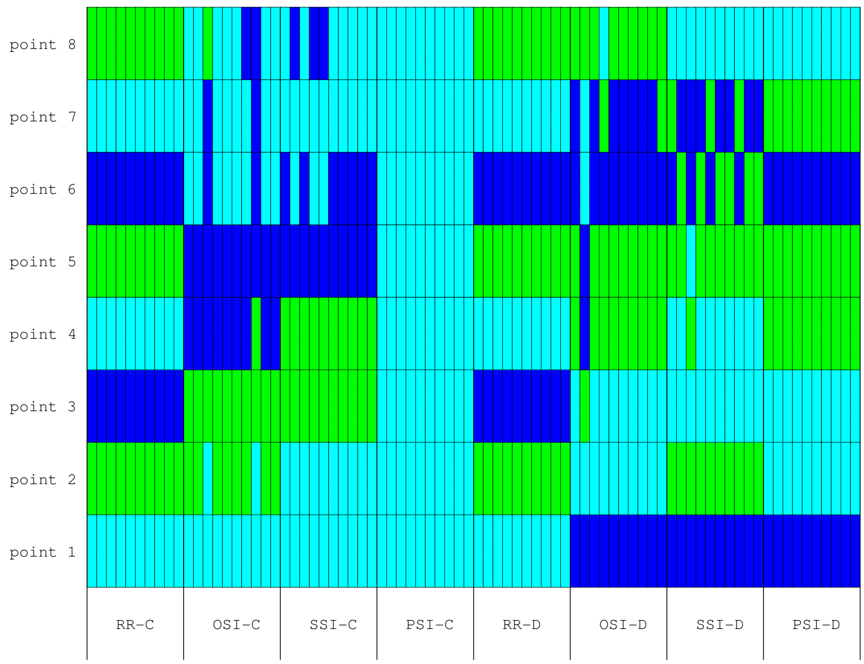
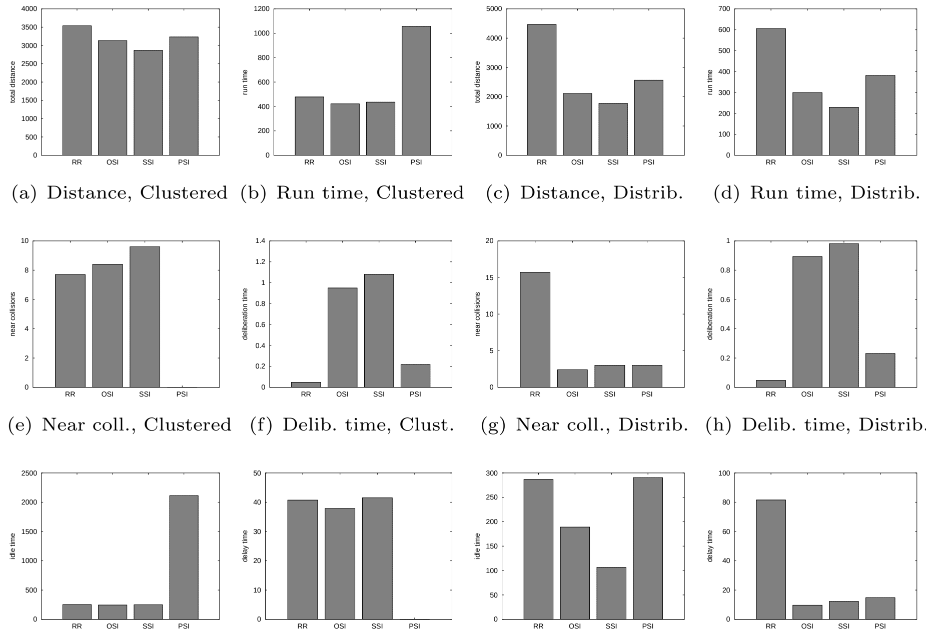


Fig. 3. The allocation of target points from set A to robots over all runs.

The approach we call the *ordered single item auction* (OSI) takes a simple step to improve on round robin allocation. In this case the target points are again placed in an ordered list, but this time each point in turn is offered to all the robots. Each robot makes a bid for the point, where the bid is the distance that the robot estimates (using an A* path planner) it will have to travel to reach the point from its current location. The point is allocated to the robot that makes the lowest bid, that robot updates its location to be the location of the target point it just acquired, and the next point is auctioned.

In the *sequential single item auction* (SSI) [7], all the target points are presented to all the robots simultaneously. Each robot bids on the target point with the lowest cost (again computed as the A* distance to the point) and the point with the lowest bid is allocated to the robot that made the bid. The winning robot updates its location to that of the target point it just won and the remaining points are re-auctioned. The process is repeated until all points have been allocated.

The *parallel single item auction* (PSI) [7] starts like SSI with all robots bidding on all points, but rather than just allocating the target point with the lowest overall bid, all the target points are allocated in one round, with each point going to whichever robot made the lowest bid on it.



(a) Distance, Clustered (b) Run time, Clustered (c) Distance, Distrib. (d) Run time, Distrib.

(e) Near coll., Clustered (f) Delib. time, Clust. (g) Near coll., Distrib. (h) Delib. time, Distrib.

(i) Idle time, Clustered (j) Delay time, Clust. (k) Idle time, Distrib. (l) Delay time, Distrib.

Fig. 4. Team results for target set A, with start points clustered (left) and distributed (right).

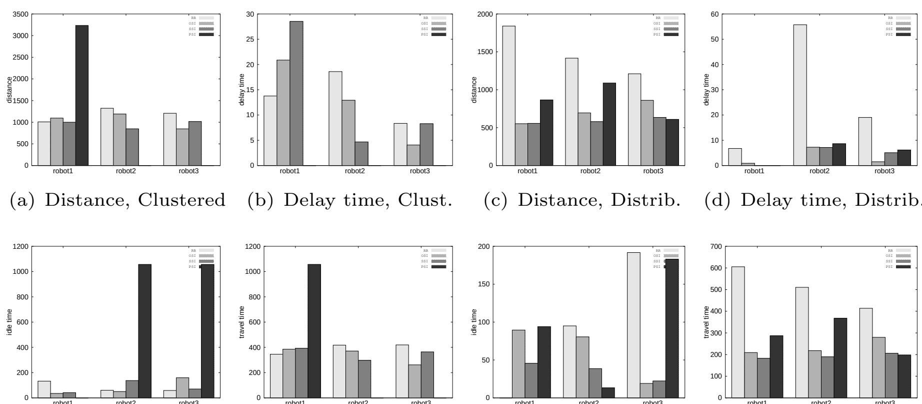
3.3 Results

The results of the experiments can be seen in Figures 2–5. Figure 2 shows the paths taken by the robots in individual runs across different scenarios as solved by different allocation mechanisms. The routes taken by different robots are given in different colors. The aim of the figure is to give a sense of allocations produced by the different mechanisms and the effect that these have on the routes taken by the robots. These traces capture many of the key points about the allocations that are echoed in the metrics. The randomness of the RR allocation mechanism is clear in the lack of a clear pattern in the allocations to each robot. The tendency of PSI to allocate points unevenly between robots — first noted in [7] — is clear when comparing it with other mechanisms for the clustered start points. The propensity for SSI to produce tight groupings is clear in its handling of the distributed start points. Note that we only show results for target points A. The allocation of target points to robots is also shown in Figure 3, revealing the patterns in each allocation mechanism. Also, note that each robot visited target points in the order in which they were allocated, which wasn't necessarily the shortest path amongst the complete set of points allocated to the robot.

Figure 4 then plots the average values over 10 runs of each metric for each of the four mechanisms on target points covering both the clustered and dis-

		Distance	Run time	Delib. time	Idle time	Delay time
Clustered	RR	3539 ± 95	478 ± 82	0.05 ± 0.0005	251 ± 52	41 ± 27
	OSI	3133 ± 95	421 ± 18	0.95 ± 0.06	245 ± 56	38 ± 23
	SSI	2865 ± 34	435 ± 86	1.08 ± 0.06	249 ± 106	42 ± 23
	PSI	3233 ± 207	1056 ± 82	0.22 ± 0.017	2113 ± 164	0 ± 0
Distributed	RR	4468 ± 168	605 ± 28	0.05 ± 0.003	286 ± 84	82 ± 30
	OSI	2108 ± 142	300 ± 27	0.89 ± 0.01	189 ± 46	10 ± 8
	SSI	1772 ± 35	229 ± 25	0.98 ± 0.043	107 ± 63	12 ± 8
	PSI	2564 ± 123	381 ± 39	0.23 ± 0.015	290 ± 71	15 ± 10

Table 1. Team metrics for target set A. Time is in seconds. Distance is in centimeters. The values given are means with 95% confidence intervals.



(a) Distance, Clustered (b) Delay time, Clust. (c) Distance, Distrib. (d) Delay time, Distrib.

(e) Idle time, Clustered (f) Travel time, Clust. (g) Idle time, Distrib. (h) Travel time, Distrib.

Fig. 5. Results for each robot on target set A with start points clustered (left) and distributed (right). The four bars for each robot are the results for the different allocation mechanisms. The bars are ordered RR, OSI, SSI and PSI.

tributed start points. Again these are only the results for target set A. Looking at the results from the two sets of start points side by side makes it clear that the clustered set of start points provides a steeper challenge for the allocation mechanism than the distributed set. While deliberation times are comparable, distance and run time are much reduced for the distributed case, as are delay times and the number of collisions (naturally each of these pairs of metrics are going to be strongly correlated). Table 1 gives the same information as Figure 4 but in tabular form, making numerical comparisons possible. The table also gives 95% confidence intervals for the metrics.

Finally, Figure 5 gives the metrics that are computed on a per robot basis, as opposed to those computed for the team as a whole. These individual metrics are distance, delay time, idle time and travel time, and the figure gives these for each of the three robots for each mechanism and both sets of start points. (Again we just show the results for target set A). Since the same robot starts in the same position each time, the distribution across the robots tells us something about the mechanisms. For example, they expose the skewed nature of the results for PSI in the case of the clustered start points, with two robots travelling no distance and reporting high idle time.

We don't give the results for the runs using the C configuration of target points since these are qualitatively very similar to those for the A points. In the next section we highlight the most interesting of the results.

3.4 Analysis

The aim of these experiments is to understand the comparative performance of the allocation mechanisms across the different scenarios with the aim of establishing the suitability of the mechanisms for different kinds of multi-robot routing tasks. As a result, our analysis focuses on the comparative performance of the mechanisms. We start by considering the results for the A target points.

Overall our analysis supports the results in [3–7], showing the effectiveness of the sequential single-item auction in finding solutions to the multi-robot routing problem when the overall distance covered is the most important performance metric. For both the clustered and distributed start points, SSI generated solutions which required the team to travel the smallest overall combined distance on average. This means that the solutions generated by SSI were executed quickly in comparison to those generated by the other allocation mechanisms, though on average in the clustered case the SSI solutions take marginally longer to execute than the OSI allocations.

The cost for this performance can be seen in the deliberation times. SSI, which requires bids from all robots for all unallocated points on all rounds, involves much more bidding than any of the other approaches, and this translates into the longest time spent in the allocation process (deliberation time). However, for the scenarios we consider here, the deliberation times are all less than 1 percent of the total task to execute the set of tasks. We return to the consideration of deliberation time in Section 4.

SSI also performs well in terms of idle time. An individual robot accumulates idle time when it finishes visiting its allocated target points before other robots finish visiting theirs, so across the team it is a measure of wasted resource. For clustered start points, SSI narrowly outperforms OSI and RR (though these results are dominated by the terrible performance of PSI on this metric), and for the distributed start points, SSI has, on average, about half the idle time of the second-best performing mechanism.

Indeed, in terms of the metrics we assess, SSI can only be considered to have poor comparative performance in terms of near collisions in the case of clustered start points. This suggests that the task allocations created by SSI in this case require the robots to traverse the same space more frequently than other allocation mechanisms (since near collisions occur when robots are trying to move past each other in close proximity). However, the cost of near collisions is increased delay time, and this is not much higher for SSI than the other mechanisms. In addition, the difference is two orders of magnitude below the run time, so is not a significant factor in the terms of task completion.

As [7] points out, PSI can come up with arbitrarily poor allocations because it does not take synergies between target points into account. As the results for the clustered start points show, it can also skew the distribution of tasks between robots — Figures 2 and 5 reveal that in the clustered start case, it allocates all the target points to one robot. This skew means that although PSI is not much worse than SSI or OSI on overall distance, it is much worse on run time and on idle time.[6] (Naturally, since it only allocates task points to one robot, there are no near collisions and hence no delay time.) Even on the distributed start points where this skew does not occur — as Figure 5 shows — PSI performs rather poorly, where overall distance travelled is more than 40% greater than that travelled, on average, in a SSI allocation.

Turning briefly to the results for the C target points, we find that they largely follow the same pattern. SSI again produces good results, broadly outperforming the other mechanisms for the clustered start points on all metrics except deliberation time. PSI however, performs considerably better when the start points are distributed (it again produces very skewed results when they are clustered). In particular, the run-time for PSI (377 seconds) is basically identical to that for SSI (375 seconds) with a considerably lower idle time (127 versus 424 seconds).

4 Discussion

Overall, our results tend to confirm the strong performance of the SSI auction mechanism in multi-robot routing tasks. The one area in which SSI performs worse than other mechanisms is in deliberation time, the time that it takes to allocate tasks to robots. For the scenarios we consider here, the cost of allocating the tasks is negligible, with the deliberation time being less than 1 percent of the total time for completing the set of tasks. The total number of bids to allocate m tasks to n robots is[7]:

$$n\left(\frac{m(m+1)}{2}\right)$$

and it is conceivable that this could become big enough to be problematic. For example, consider the case of one hundred robots — as in the Centibots project [14] — which have to allocate 500 target points. In such a case the SSI auction would require over 12 million bids, a 12,000-fold increase over what is required in our experiments, and enough to make deliberation time a significant contributor to the time for task completion. In addition, since each bid has to be transmitted wirelessly — either to a centralised auction manager, or to all

[6] The distance for PSI is 13% larger than that for OSI and 23% larger than that for SSI, but the runtime for PSI is 2.4 times that for SSI and 2.5 times that for OSI.

[7] One bid from each robot for m tasks in the first round, one bid from each robot for $m-1$ tasks in the second round, and so on.

other robots in a distributed auction — the number of messages can be a factor in robot deployments where communication bandwidth is limited [15].

Both of these aspects might make the OSI or PSI mechanisms worth considering for larger deployments. The OSI auction results in distances that are 10% (clustered start) to 20% (distributed start) worse than SSI in terms of distance to the A target points, and for our Centibots example would require several hundred times fewer bids (50,000 rather than 12 million). The PSI auction is about 40% worse in terms of total distance for the distributed case with the A target points, but could do the task allocation for the Centibots with just 500 bids (one for each task). In the clustered case with the A target points, as we pointed out above, PSI is only about 23% worse than SSI in terms of distance travelled, but the long run time that results from the skewed allocation needs to be taken into account. This point is reinforced by the results for the C target points, which show that there are situations in which PSI can equal or outperform SSI on some metrics. However, while our results are encouraging in this respect, an analysis of more scenarios would be required before we can reach any firm conclusions about whether PSI can predictably outperform SSI in a way that would allow one to select the best mechanism based on the characteristics of the problem being solved.

5 Related work

As mentioned above, there is much related work on auction mechanisms for task allocation in robotics. Some of this has already been mentioned, and here we take a look at the relevant work that we have not yet covered.

The use of market mechanisms in distributed computing can be considered to start with Smith's contract net protocol [16], and this was followed by Wellman and Wurman's [17] *market-aware agents*. Wellman and Wurman [17] point out that every decision in a MAS can be considered to be a decision about resource allocation with obvious application to multirobot systems to optimize resource usage, communication and task completion time [18].

A primary strength of market-based approaches is their reliance only on local information or the self-interest of the agents to arrive at efficient solutions to large-scale, complex problems that are otherwise intractable [2, 19]. The most common instantiations of market-based approaches in multirobot systems are auctions. Auctions are commonly used for distribution tasks, where resources or roles are treated as commodities and auctioned to agents. Based on their private preference information for a particular commodity, agents bid in these auctions. A significant body of work analyzes the effects of different auction mechanisms [16, 20–22], bidding strategies [5, 23], dynamic task re-allocation or swapping [24] and levels of commitment to the contracts [25] on the overall solution quality. The application domains vary from loosely coupled tasks, such as exploration [4, 18, 22], to tightly-coupled tasks [18, 26] such as box pushing and formation control [27], which require close coordination between robots. Auctions have also been used for role allocation in robot soccer [28] and in heterogeneous teams [29].

In domains where there is a strong synergy between items, single-item auctions can result in sub-optimal allocations [20]. In the domain of interest here, multi-robot exploration, a strong synergy exists between target points that robots need to explore. Combinatorial auctions emedy this limitation by allowing agents to bid on bundles of items, and minimize the total travel distance because they take the synergies between target points into account [6]. Combinatorial auctions suffer, however, from the computational costs of bid generation and bundle valuation by the agents, and winner determination by the auction mechanism itself, all of which are NP-hard [7].

Of all the literature on auctions in multi-robot teams, [30] and [31] are the most closely related to our work. Both evaluate SSI in simulation and so could argue their work evaluates the practical cost of solutions generated using SSI (and [30] provides some results from real robots). However, the focus of both [30] and [31] is on finding optimal mechanisms for dynamic task reallocation during execution rather than, like our work, attempting to characterise auction mechanisms in terms of the tasks for which they are best fitted. In addition, neither consider the range of metrics that we do and so are unable, for example to comment on the load balancing that measuring idle time exposes.

6 Summary

This paper has studied the performance of a number of auction mechanisms on a version of the multi-robot routing problem. It has gone further than any work we are aware of in testing auction mechanisms across a range of performance metrics and in a way that tests how the mechanisms would work on a fleet of real robots.

The main result is that the sequential single item (SSI) auction broadly outperforms other single item auctions across our range of metrics, though it does not perform best on all of them for all scenarios. However, there do seem to be trade-offs, especially in terms of the total number of bids required by the mechanisms, which suggests that the SSI auction might have issues with scaling to larger routing problems than we study here, especially if communication bandwidth is restricted. In other words, the high performance of SSI comes at a cost that might be hard to pay for some scenarios. Other mechanisms we tested, which can scale better, might be preferable on such scenarios despite their poorer performance.

As [7] points out, SSI allocations can be improved by moving to combinatorial auctions that can exploit synergies between tasks by having robots bid on bundles of target points. However, as [7] also shows, combinatorial auctions require more bids (since there are more bundles than individual items). As we pointed out above, winner determination in combinatorial auctions is also a computationally hard problem. This suggests that the trade-off between deliberation time and run time will be even sharper for combinatorial auctions than it is for SSI auctions, and this is a topic that we plan to study next.

Acknowledgements

This work was partially funded by the National Science Foundation (NSF) under grants #IIS-1116843 and #IIS-1338884, by a University of Liverpool Research Fellowship and by a Fulbright-King's College London Scholar Award.

References

1. Koenig, S., Tovey, C.A., Zheng, X., Sungur, I.: Sequential bundle-bid single-sale auction algorithms for decentralized control. In: Proceedings of the International Joint Conference on Artificial Intelligence. (2007)
2. Dias, M.B., Zlot, R., Kalra, N., Stentz, A.: Market-based multirobot coordination: A survey and analysis. Technical Report CMU-RI-TR-05-13, Carnegie Mellon University (2005)
3. Tovey, C., Lagoudakis, M.G., Jain, S., Koenig, S.: Generation of Bidding Rules for Auction-Based Robot Coordination. In: Proceedings of the 3rd International Multi-Robot Systems Workshop. (March 2005)
4. Lagoudakis, M., Berhault, M., Koenig, S., Keskinocak, P., Kelywegt, A.: Simple auctions with performance guarantees for multi-robot task allocation. In: Proceedings of the International Conference on Intelligent Robotics and Systems (IROS). (2004)
5. Lagoudakis, M., Markakis, V., Kempe, D., Keskinocak, P., Koenig, S., Kleywegt, A., Tovey, C., Meyerson, A., Jain, S.: Auction-based multi-robot routing. In: Proceedings of Robotics: Science and Systems Conference. (2005)
6. Koenig, S., Keskinocak, P., Tovey, C.: Progress on agent coordination with cooperative auctions. In: Proceedings of the AAAI Conference on Artificial Intelligence. (2010)
7. Koenig, S., Tovey, C., Lagoudakis, M., Kempe, D., Keskinocak, P., Kleywegt, A., Meyerson, A., Jain, S.: The Power of Sequential Single-Item Auctions for Agent Coordination. In: Proceedings of National Conference on Artificial Intelligence. (2006)
8. Sklar, E.I., Özgelen, A.T., Muñoz, J.P., Gonzalez, J., Manashirov, M., Epstein, S.L., Parsons, S.: Designing the HRTeam Framework: Lessons Learned from a Rough-and-Ready Human/Multi-Robot Team. In: Proceedings of the Workshop on Autonomous Robots and Multirobot Systems (ARMS) at Autonomous Agents and MultiAgent Systems (AAMAS), Taipei, Taiwan (May 2011)
9. Özgelen, A.T., Schneider, E., Sklar, E.I., Costantino, M., Epstein, S.L., Parsons, S.: A first step toward testing multiagent coordination mechanisms on multi-robot teams. In: Proceedings of the Workshop on Autonomous Robots and Multirobot Systems (ARMS) at Autonomous Agents and MultiAgent Systems (AAMAS). (2013)
10. Ezhilchelvan, P., Morgan, G.: A dependable distributed auction system: Architecture and an implementation framework. In: Proceedings of the 5th International Symposium on Autonomous Decentralized Systems. (2001)

11. Gerkey, B., Vaughan, R.T., Howard, A.: The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems. In: Proceedings of the 11th International Conference on Advanced Robotics. (2003)
12. Habib, M.K.: Humanitarian Demining: Reality and the Challenge of Technology. International Journal of Advanced Robotic Systems **4**(2) (2007)
13. Buehler, M., Iagnemma, K., Singh, S.: The 2005 DARPA Grand Challenge: The great robot race. Springer (2007)
14. Konolige, K., Fox, D., Ortiz, C., Agno, A., Eriksen, M., Limketkai, B., Ko, J., Morisset, B., Schulz, D., Stewart, B., et al.: Centibots: Very large scale distributed robotic teams. In: Experimental Robotics IX. Springer (2006) 131–140
15. Savkin, A.V.: The problem of coordination and consensus achievement in groups of autonomous mobile robots with limited communication. Nonlinear Analysis: Theory, Methods & Applications **65**(5) (2006) 1094–1102
16. Smith, R.G.: The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. In Bond, A.H., Gasser, L., eds.: Distributed Artificial Intelligence. Morgan Kaufmann Publishers Inc. (1988)
17. Wellman, M.P., Wurman, P.R.: Market-Aware Agents for a Multiagent World. Robotics and Autonomous Systems **24** (1998) 115–125
18. Gerkey, B.P., Mataríc, M.J.: Sold!: Auction methods for multi-robot control. IEEE Transactions on Robotics and Automation Special Issue on Multi-Robot Systems **18**(5) (2002)
19. Dias, M.B., Stentz, A.: A Free Market Architecture for Distributed Control of a Multirobot System. In: Proceedings of the 6th International Conference on Intelligent Autonomous Systems. (July 2000)
20. Berhault, M., Huang, H., Keskinocak, P., Koenig, S., Elmaghraby, W., Griffin, P.M., Kleywegt, A.: Robot Exploration with Combinatorial Auctions. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. (2003)
21. Kraus, S.: Automated negotiation and decision making in multiagent environments. In: MultiAgent Systems and Applications. Springer-Verlag (2001)
22. Zlot, R., Stentz, A., Dias, M.B., Thayer, S.: Multi-robot exploration controlled by a market economy. In: Proceedings of the IEEE Conference on Robotics and Automation. (2002)
23. Sariel, S., Balch, T.: Efficient bids on task allocation for multi-robot exploration. In: Proceedings of the Nineteenth Florida Artificial Intelligence Research Society Conference. (2006)
24. Golfarelli, M., Maio, D., Rizzi, S.: A Task-Swap Negotiation Protocol Based on the Contract Net Paradigm. Technical Report 005-97, DEIS, CSITE - Università di Bologna (1997)
25. Mataric, M., Sukhatme, G., Ostergaard, E.: Multi-robot task allocation in uncertain environments. Autonomous Robots (2003)
26. Kalra, N., Stenz, A.: A Market Approach to Tightly-Coupled Multi-Robot Coordination: First Results. In: Proceedings of the ARL Collaborative Technologies Alliance Symposium. (2003)
27. Michael, N., Zavlanos, M.M., Kumar, V., Pappas, G.J.: Distributed multi-robot task assignment and formation control. In: Proceedings of the IEEE International Conference on Robotics and Automation. (2008)
28. Frias-Martinez, V., Sklar, E.I., Parsons, S.: Exploring auction mechanisms for role assignment in teams of autonomous robots. In: Proceedings of the 8th RoboCup International Symposium. (2004)

29. Thomas, G., Williams, A.B.: Sequential auctions for heterogeneous task allocation in multiagent routing domains. In: IEEE International Conference on Systems, Man and Cybernetics, IEEE (2009)
30. Nanjanath, M., Gini, M.: Repeated auctions for robust task execution by a robot team. Robotics and Autonomous Systems **58**(7) (2010) 900–909
31. Schoenig, A., Pagnucco, M.: Evaluating sequential single-item auctions for dynamic task allocation. In: AI 2010: Advances in Artificial Intelligence. Springer (2011)